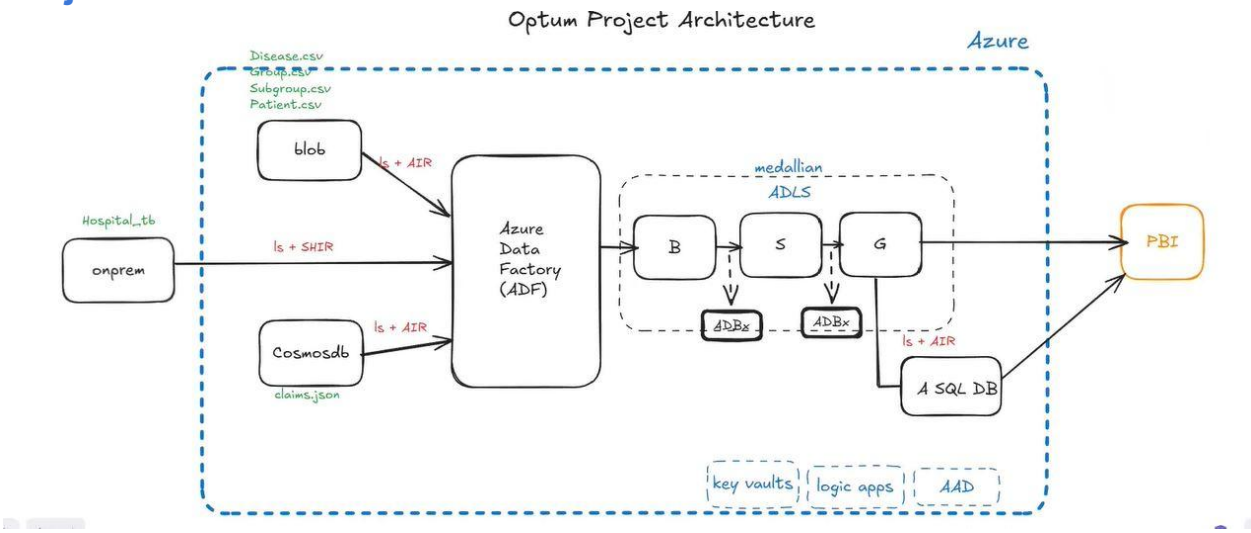
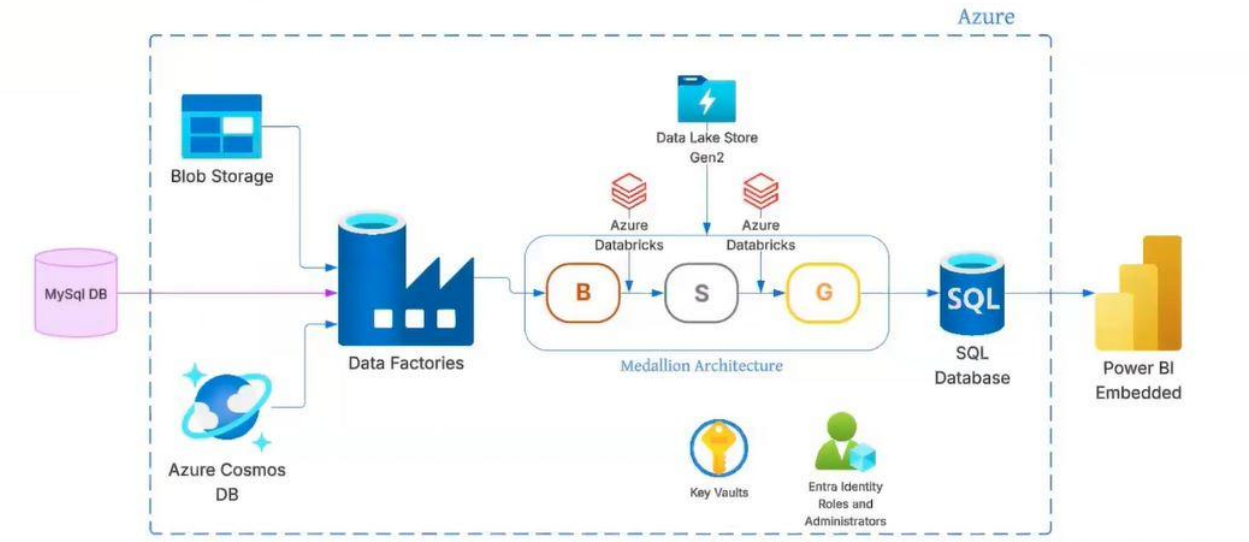
Optum Project

Project Objective:

This project aims to centralize data from multiple sources including MySQL, Azure Cosmos DB, and Blob Storage, enabling scalable and reliable data ingestion and transformation. It implements a Medallion Architecture (Bronze, Silver, and Gold) to ensure data quality, consistency, and governance.

Project Architecture:





Data Sources:

- MySql : Hospital_TB
- Azure CosmosDB : claims.json
- Azure Blob Storage : disease.csv, group.csv, subgroup.csv, patient.csv, subscriber_record.csv

Data Ingestion and Orchestration:

- Azure Data Factory

Data Processing:

- Databricks

Data Serving Layer:

- Azure Database

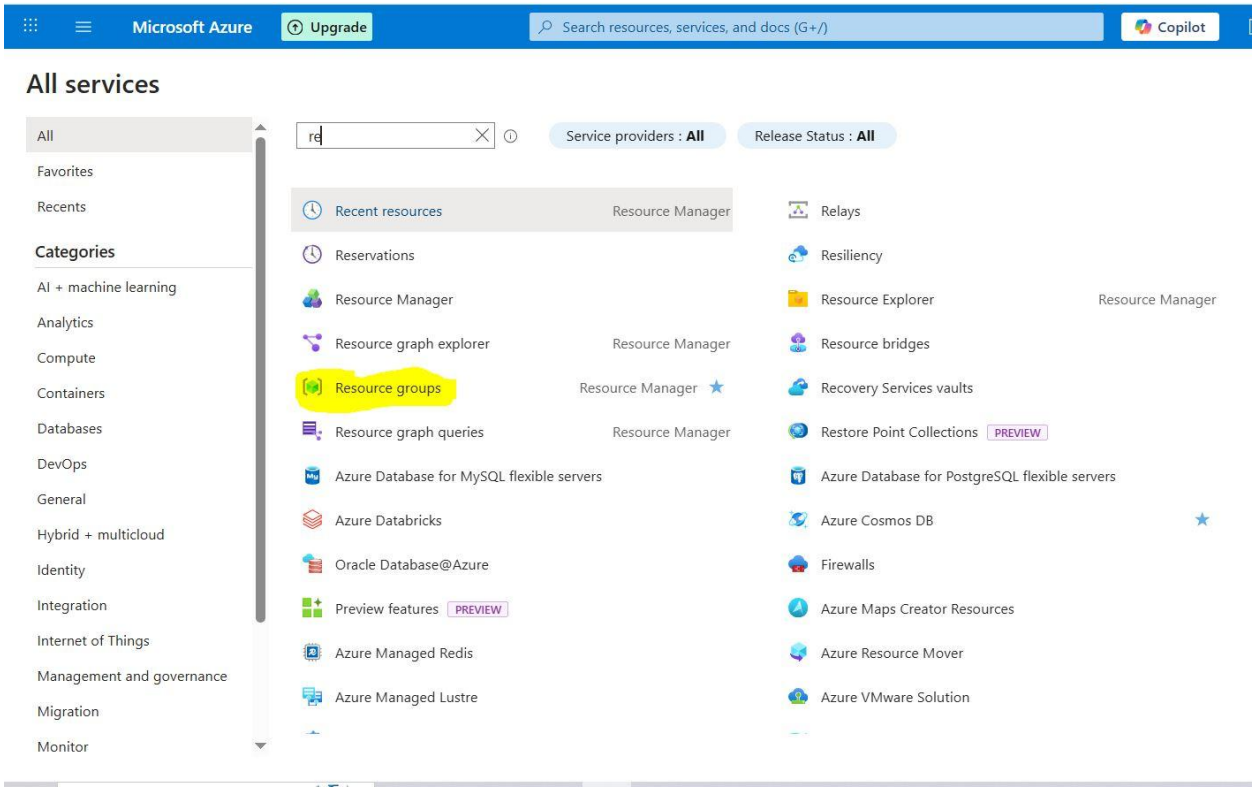
Creating services:

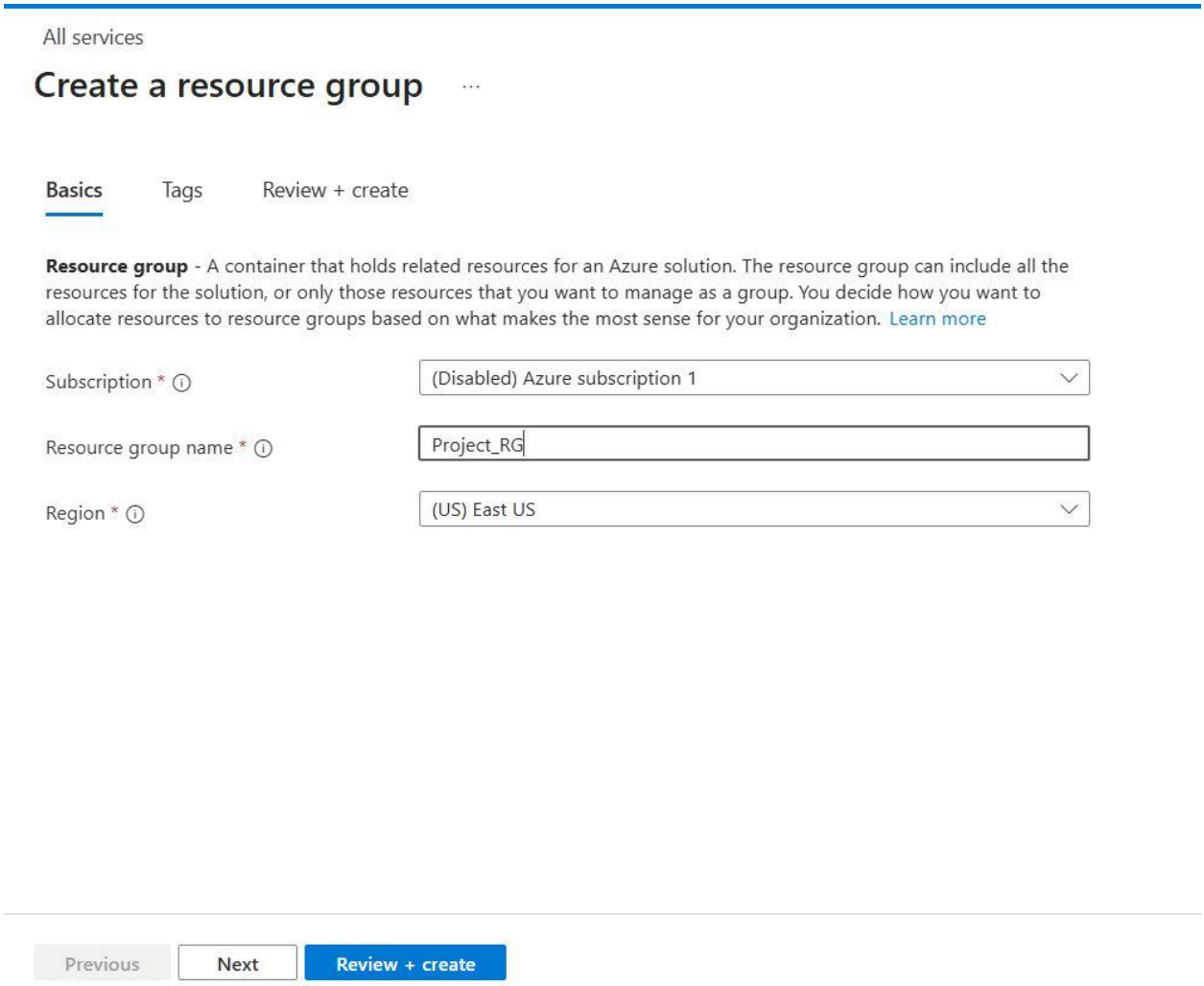
1.Resource Group

Resource group is like a Single Folder to keep all the related azure services for the particular project.

1.1.How to Create Resourse Group

In Azure, Click Resource Group and Create it.





2.Azure Blob Storage

Create Azure blob and load the datasets

Home >

Create a resource

... Design a new Azure workload | Help me build a new Azure C

Get Started

Recently created

Categories

Machine Learning

AI Apps and Agents

Analytics

Blockchain

Infrastructure Services

Databases

Developer Tools

DevOps

Identity

Integration

Internet of Things

IT & Management Tools

Media

Search services and marketplace G

Popular Azure services See more in All services

Function App
Create

Web App
Create

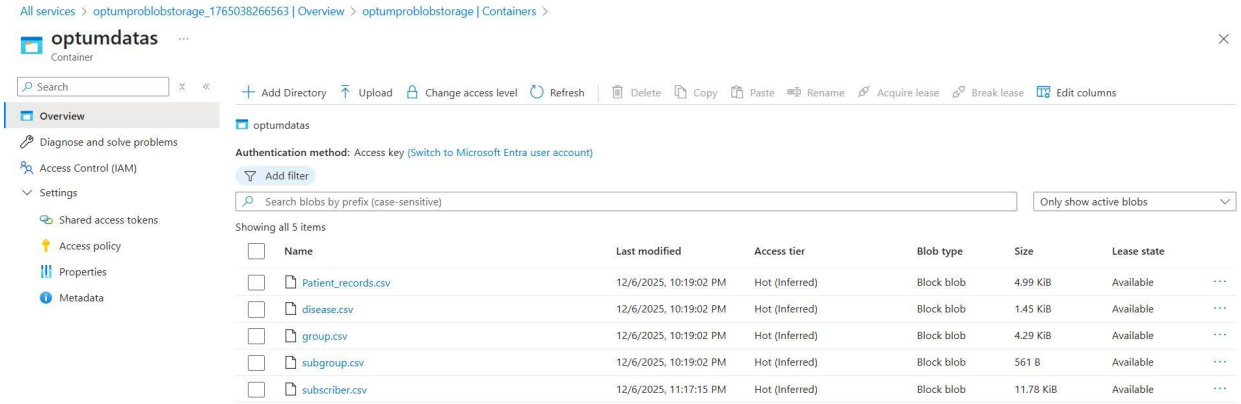
Virtual network
Create

Resource group
Create

Key Vault
Create

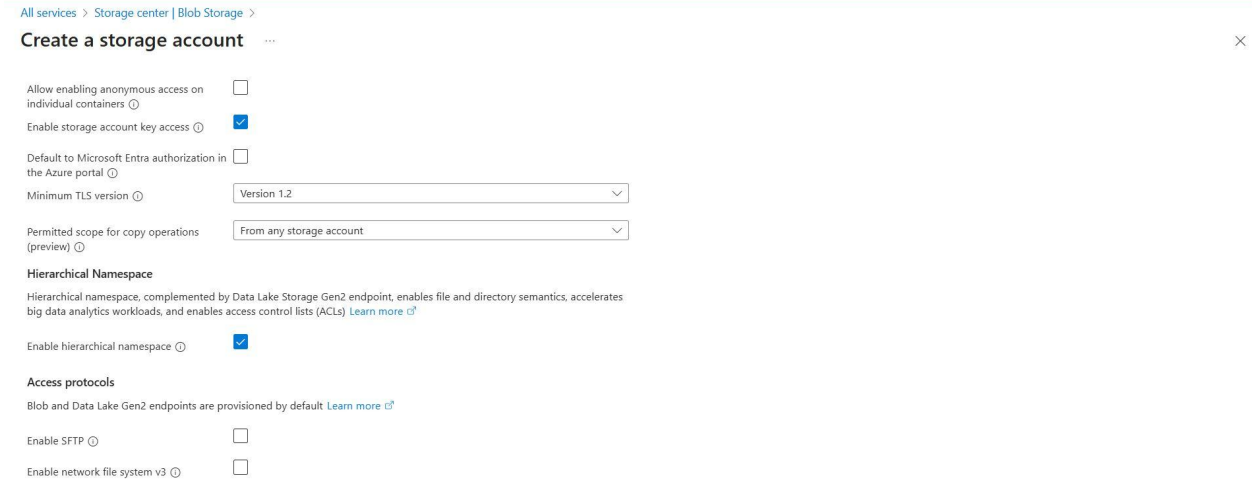
Virtual machine
Create

Storage account
Create

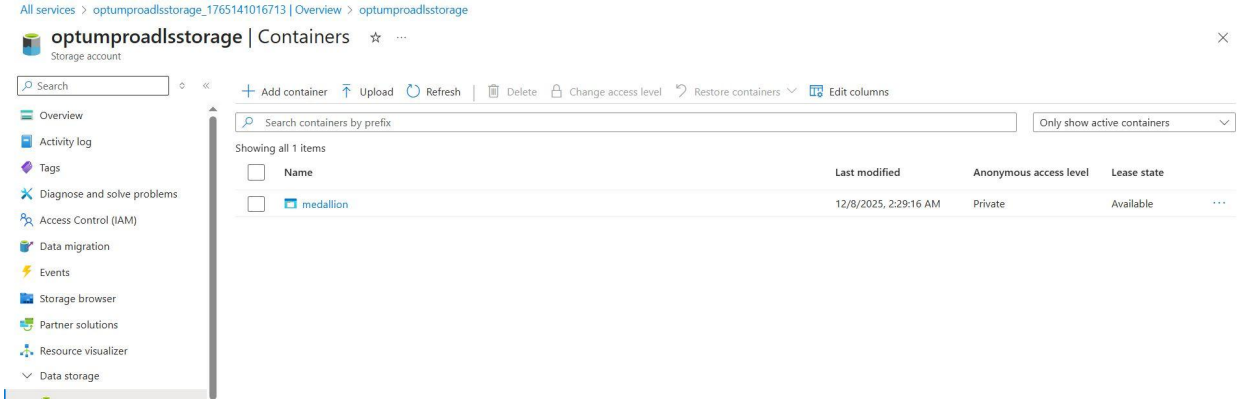


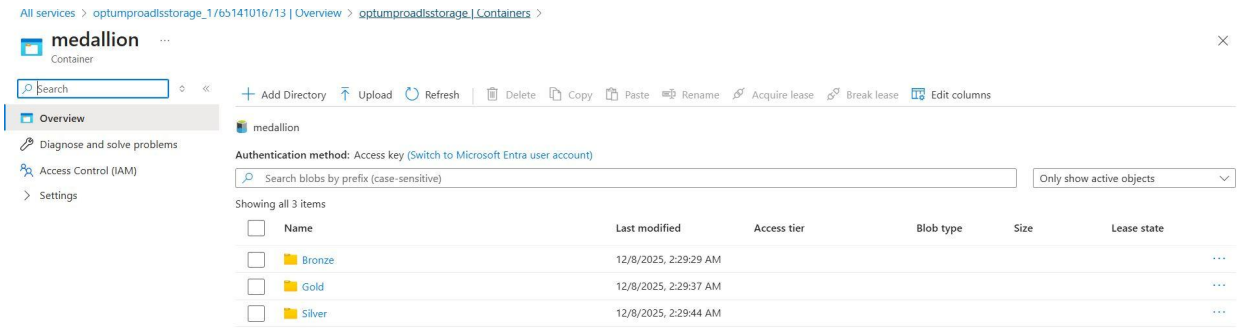
3.Azure datalake storage gen2

1.Create adls to create multiple directory for organising the data in a medallion architecture. Adls is in actual blob storage which needs to enable ‘Hierarical namespace’ while creating it.



2.Create Directories as Medallion, in medallion folder, create 3 folders, ‘Bronze’, ‘Silver’, ‘Gold’.

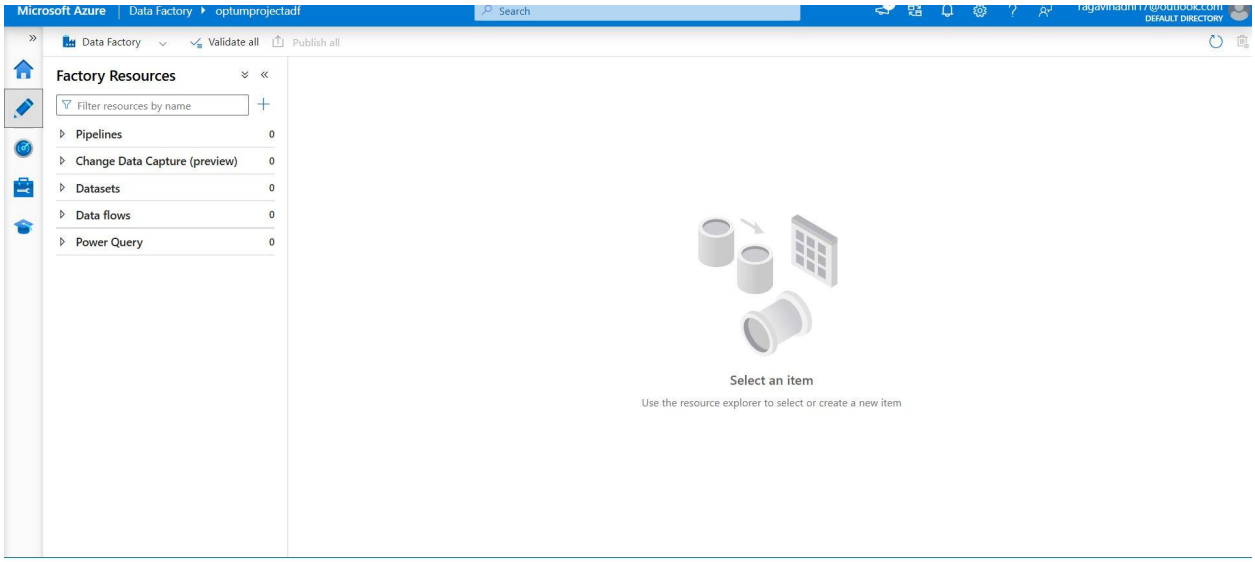


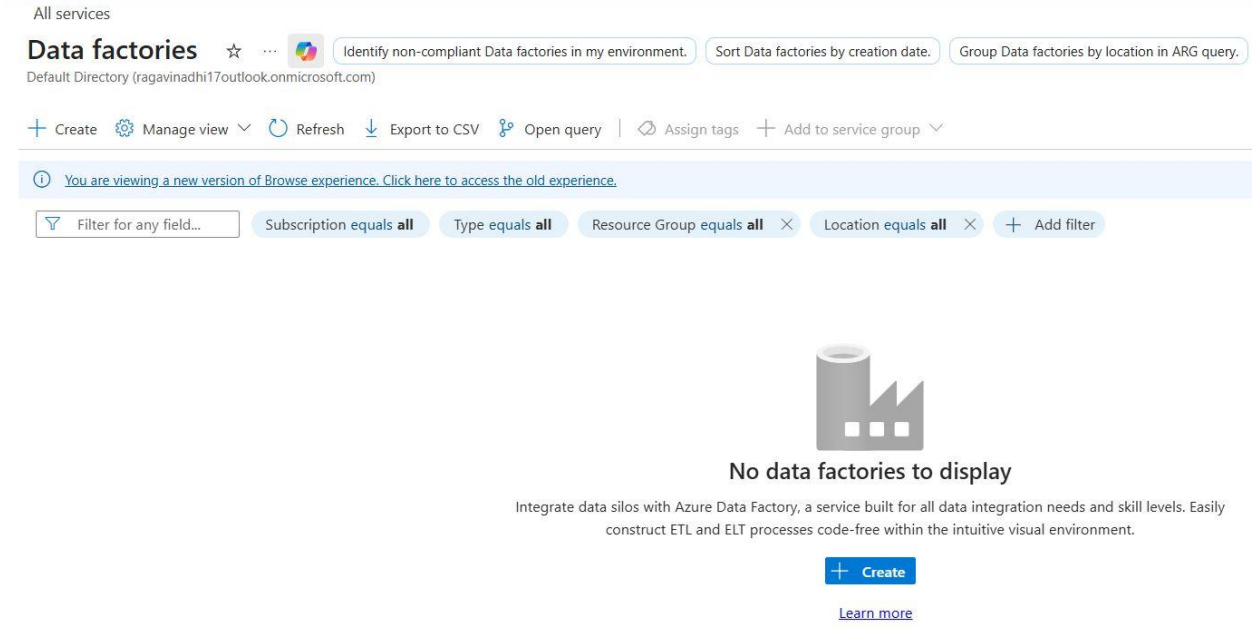


3.Azure Data Factory

1.Create adf for orchestration and data ingestion.

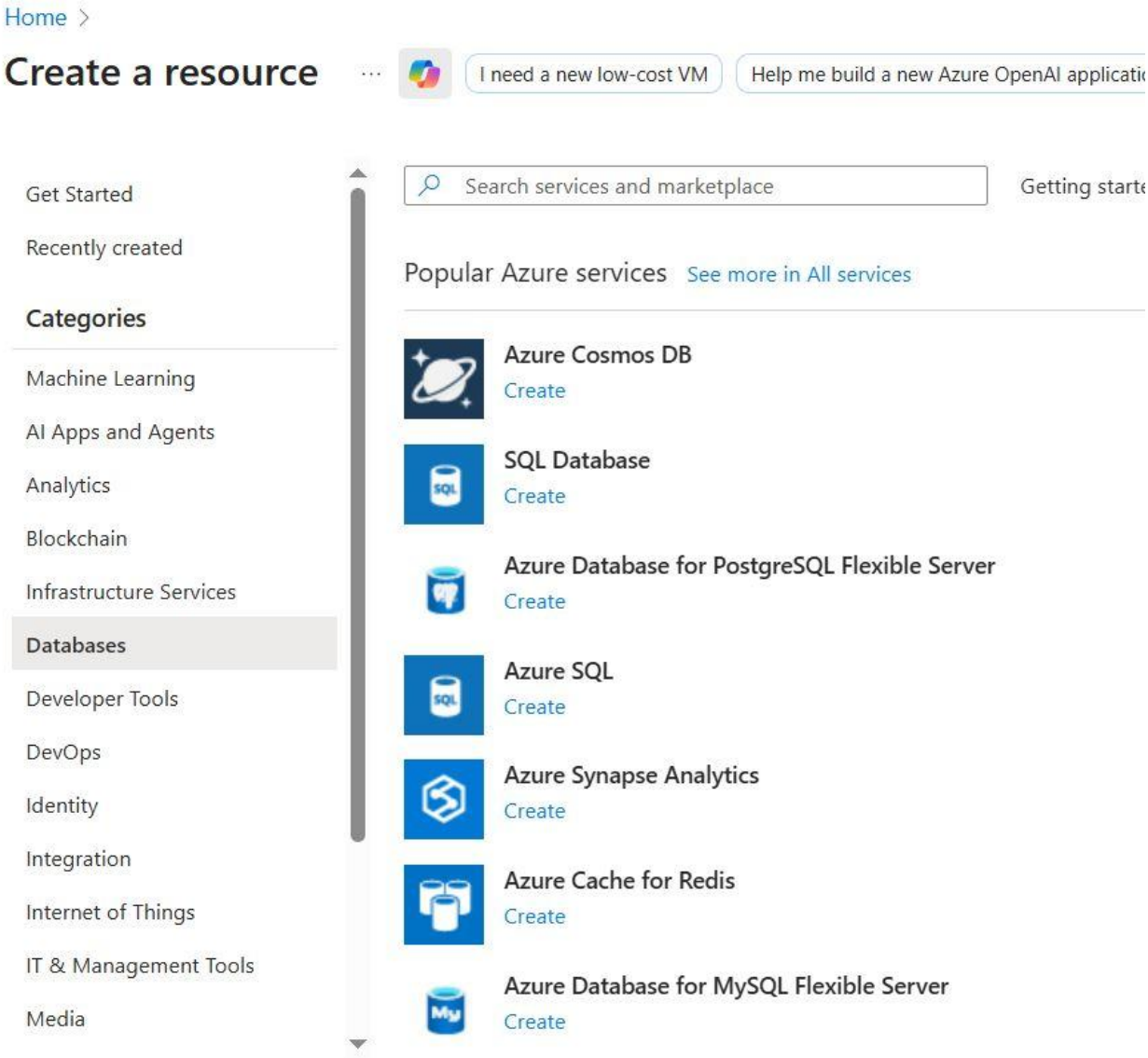
2.Create from azure services

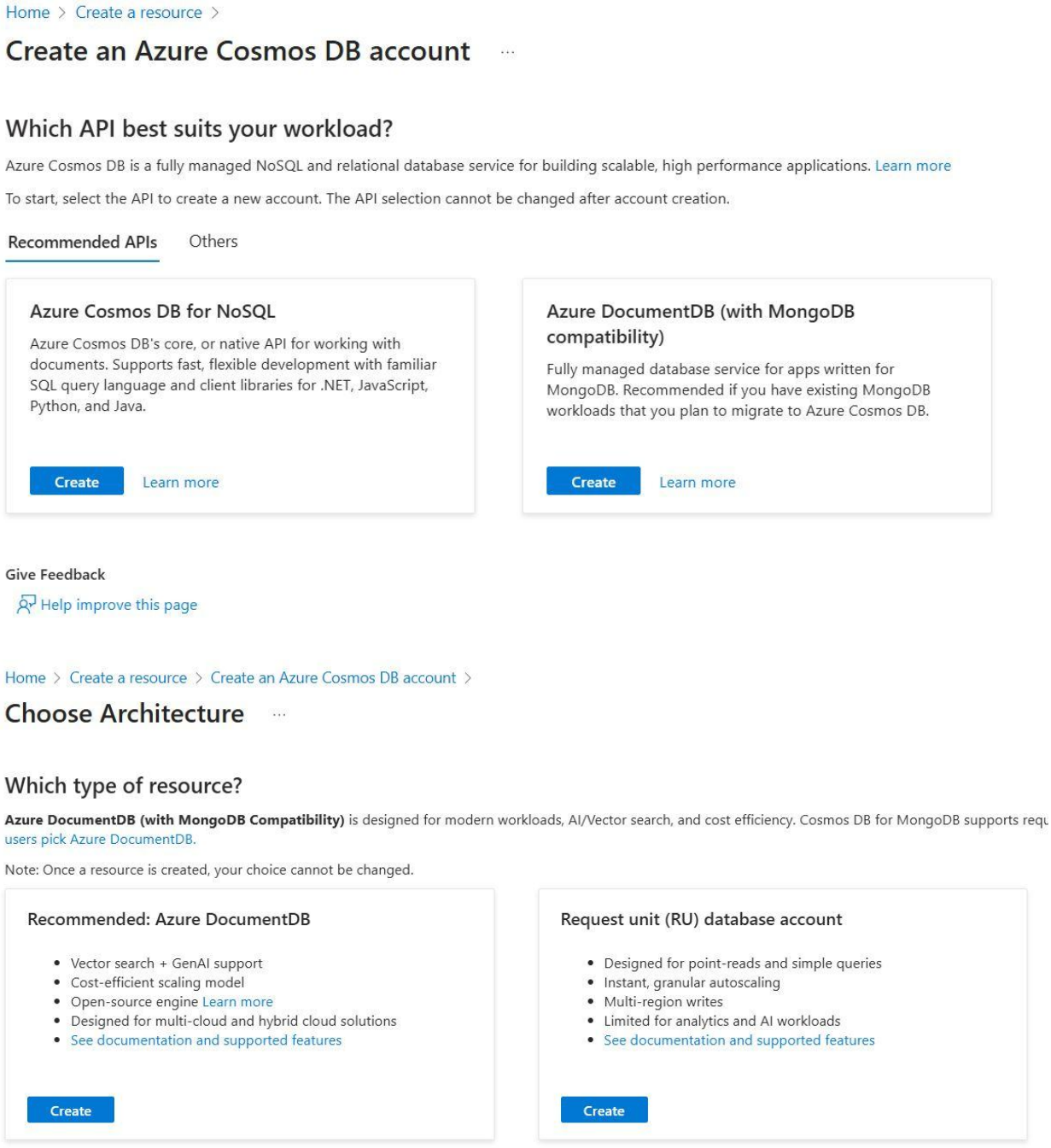




4.Azure CosmosDB

1.Create Cosmosdb from azure services
2.Select Azure cosmos db for nosql
3.select request unit(RI) database account

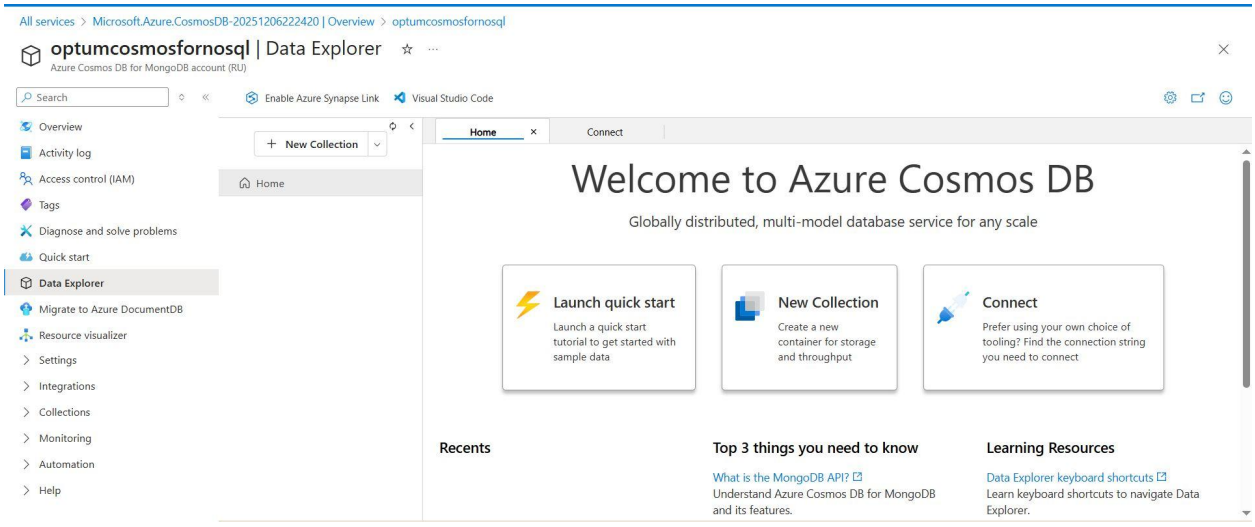


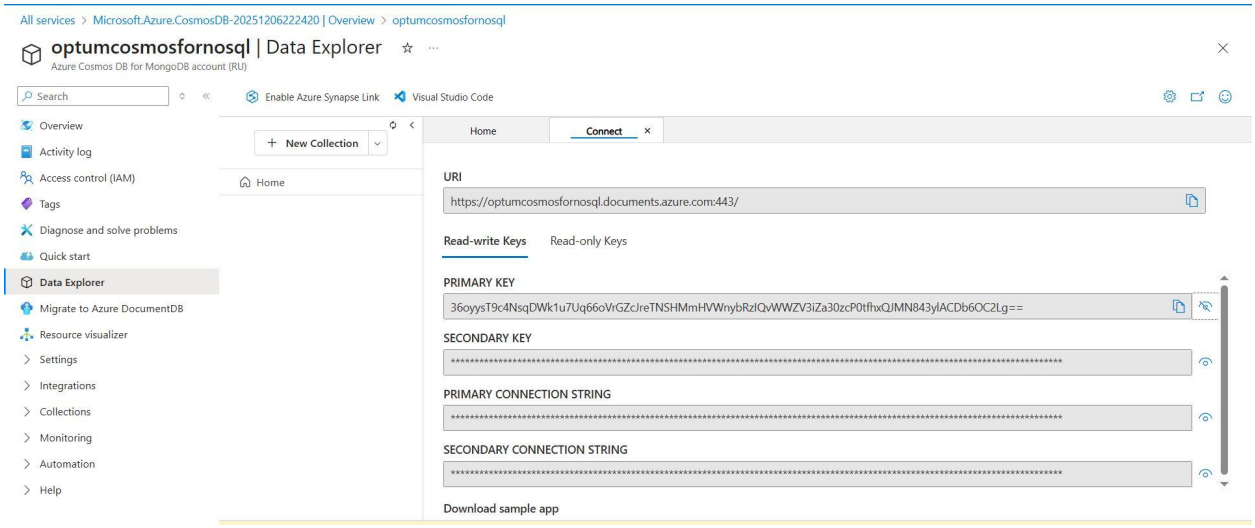


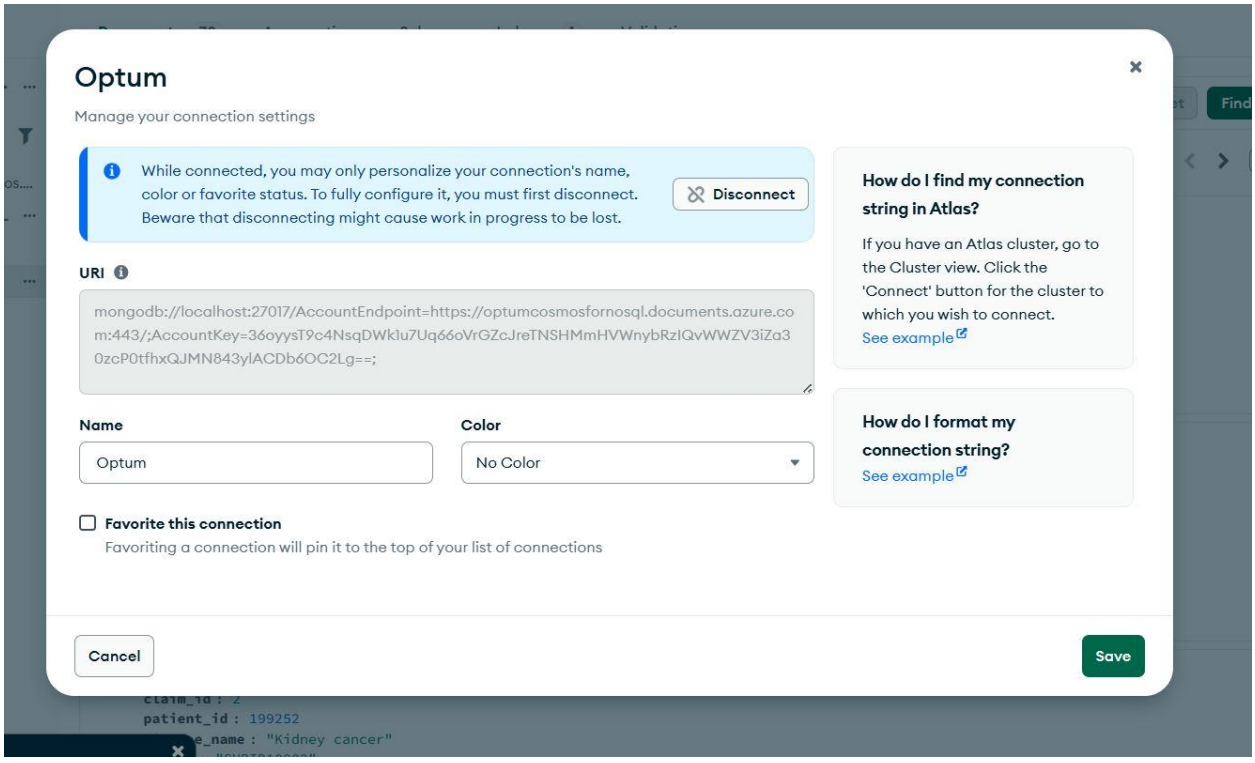
4. Open Local MongoDB and try connecting with CosmosDB with String Connections.

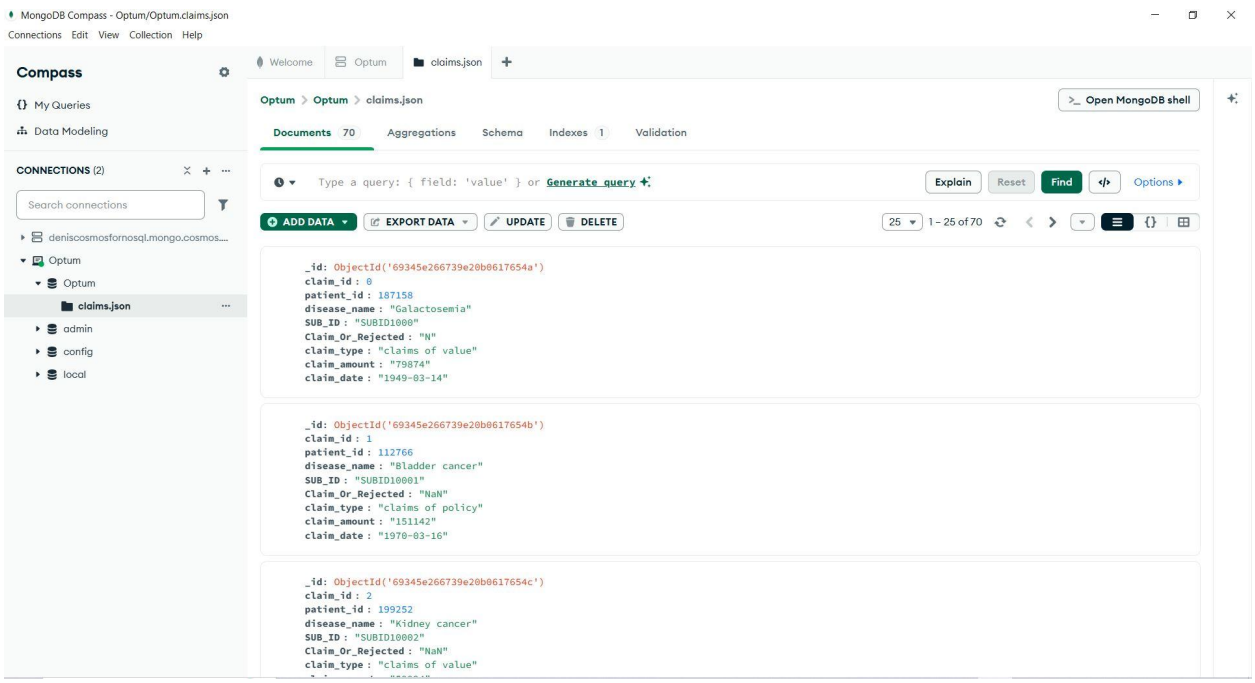
5. In data explorer, click Connect, inside it, copy ‘Primary Connection String’

6 .paste the connection string in mongodb so that, we can connect the datas from mongodb.


All services > Microsoft.Azure.CosmosDB-20251206222420 | Overview > optumcosmosfornosql
optumcosmosfornosql | Data Explorer
Azure Cosmos DB for MongoDB account (RU)
Search
Enable Azure Synapse Link
Visual Studio Code
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Quick start
Data Explorer
Migrate to Azure DocumentDB
Resource visualizer
Settings
Integrations
Collections
Monitoring
Automation
Help
New Collection
Home
Connect
Welcome to Azure Cosmos DB
Globally distributed, multi-model database service for any scale
Launch quick start
Launch a quick start tutorial to get started with sample data
New Collection
Create a new container for storage and throughput
Connect
Prefer using your own choice of tooling? Find the connection string you need to connect
Recents
Top 3 things you need to know
What is the MongoDB API?
Understand Azure Cosmos DB for MongoDB and its features.
Learning Resources
Data Explorer keyboard shortcuts
Learn keyboard shortcuts to navigate Data Explorer.

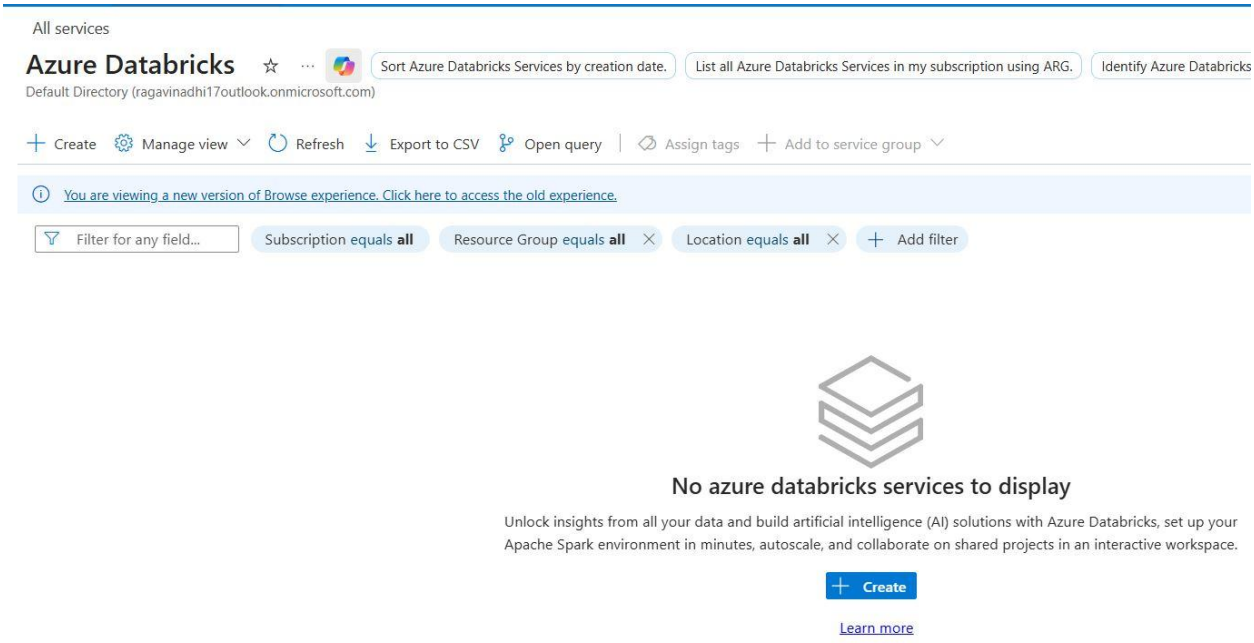

All services > Microsoft.Azure.CosmosDB-20251206222420 | Overview > optumcosmosfornosql
optumcosmosfornosql | Data Explorer
Azure Cosmos DB for MongoDB account (RU)
Search
Enable Azure Synapse Link
Visual Studio Code
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Quick start
Data Explorer
Migrate to Azure DocumentDB
Resource visualizer
Settings
Integrations
Collections
Monitoring
Automation
Help
New Collection
Home
Connect
URI
https://optumcosmosfornosql.documents.azure.com:443/
Read-write Keys
Read-only Keys
PRIMARY KEY
36oyysT9c4NsqDWk1u7Uq66oVrGZcJreTNSHMmHVWnybRzIQvWWZV3iZa30zcP0tfhxQJMN843ylACDb6OC2Lg==
SECONDARY KEY
PRIMARY CONNECTION STRING
SECONDARY CONNECTION STRING
Download sample app


Optum
Manage your connection settings
While connected, you may only personalize your connection's name, color or favorite status. To fully configure it, you must first disconnect. Beware that disconnecting might cause work in progress to be lost.
Disconnect
URI
mongodb://localhost:27017/AccountEndpoint=https://optumcosmosfornosql.documents.azure.com:443/;AccountKey=36oyysT9c4NsqDWk1u7Uq66oVrGZcJreTNSHMmHVWnybRzIQvWWZV3iZa30zcP0tfhxQJMN843ylACDb6OC2Lg==;
Name
Optum
Color
No Color
Favorite this connection
Favoriting a connection will pin it to the top of your list of connections
How do I find my connection string in Atlas?
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.
See example
How do I format my connection string?
See example
Cancel
Save



5.Azure Databricks

1.Create Databricks from azure services to process/clean/transform the datas.



6.Azure SqlDB

1.Create SQL DB from azure services which is going to be centralized database to store the final denormalized table.

Fill in the areas;

- Servername : Optumsqldb
- Location : East Asia
- Authentication method : use SQL authentication

- Password : ******
- Confirm Password : ******

2.Choose Standard or Basic Pricing Tier(for low cost)

- Backup Storage Redundancy : Locally Redundant backup Storage

3.Enable public Access(Firewall Rule):

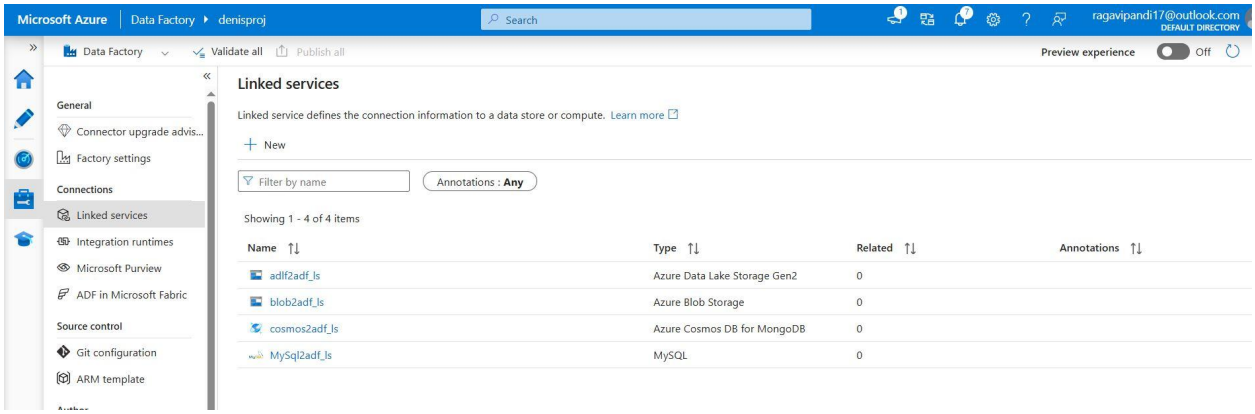
- Connectivity method : Public Endpoint

4.Review and create

Creating link services:

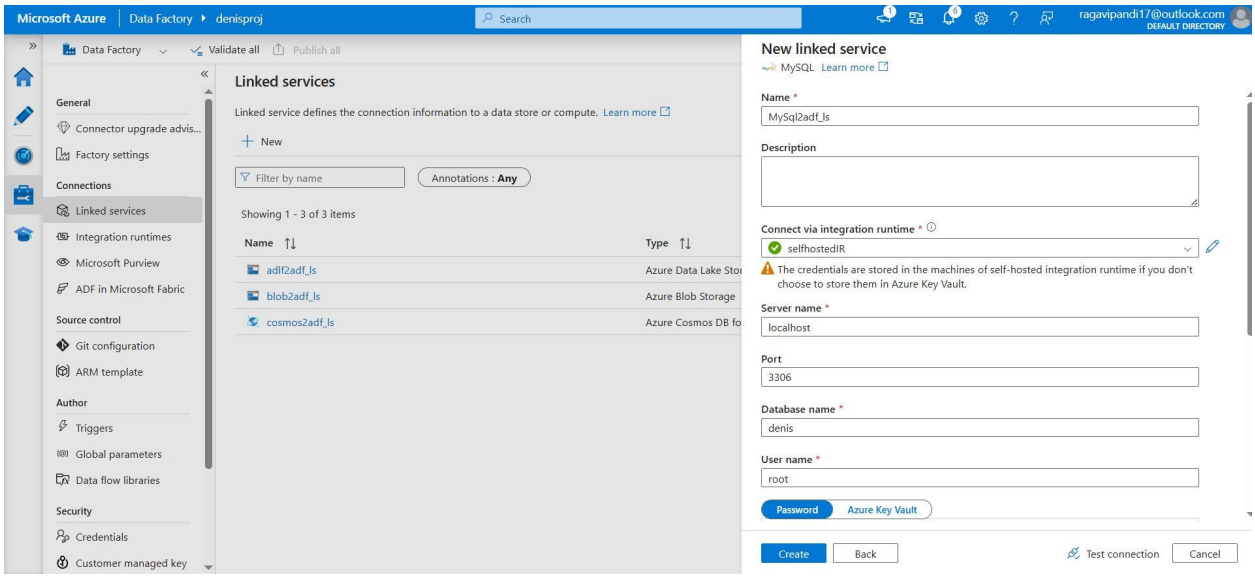
1.adls2adf_ls
2.blob2adf_ls
3.cosmosdb2adf_ls
4.mysql2adf_ls

a.Create all linked services, it is like a path for data ingestion.

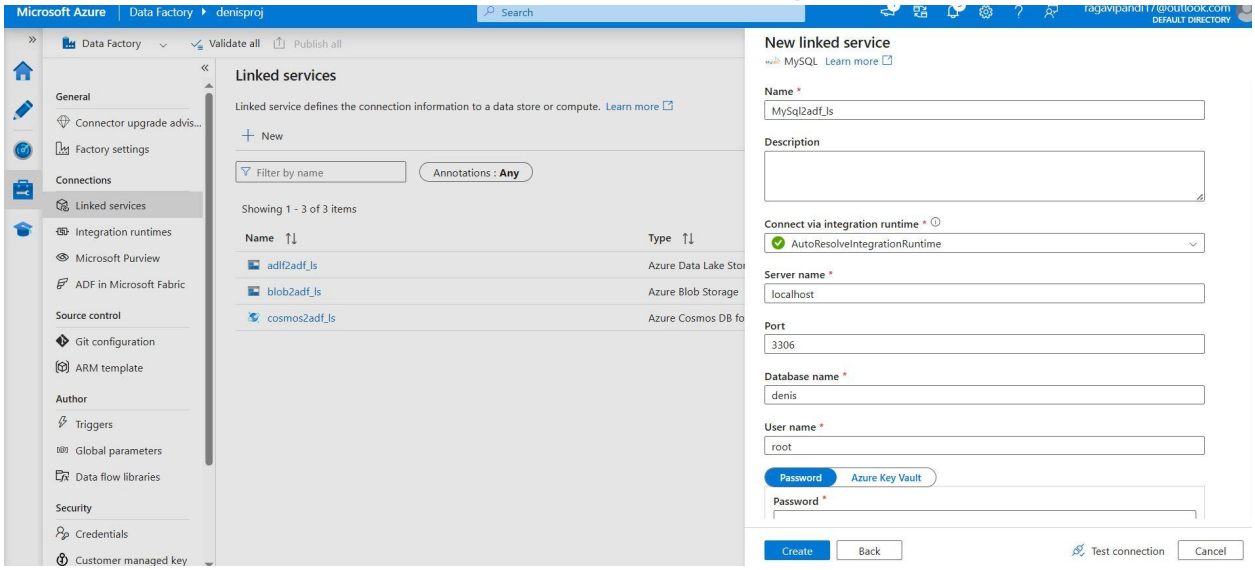


b.Before creating link service to **Onprem, 'Self hosted integrated runtime'** should be installed in onprem device.

C. connect with onprem with Mysql Credentials



D. With other services, **'Give AutoResolveIntegrationRuntime'**



Create Pipelines in ADF:

1.Raw2Bronze_PL:

Data move: Blob to ADLS

1.Create Pipeline in adf

2.Copy Activity

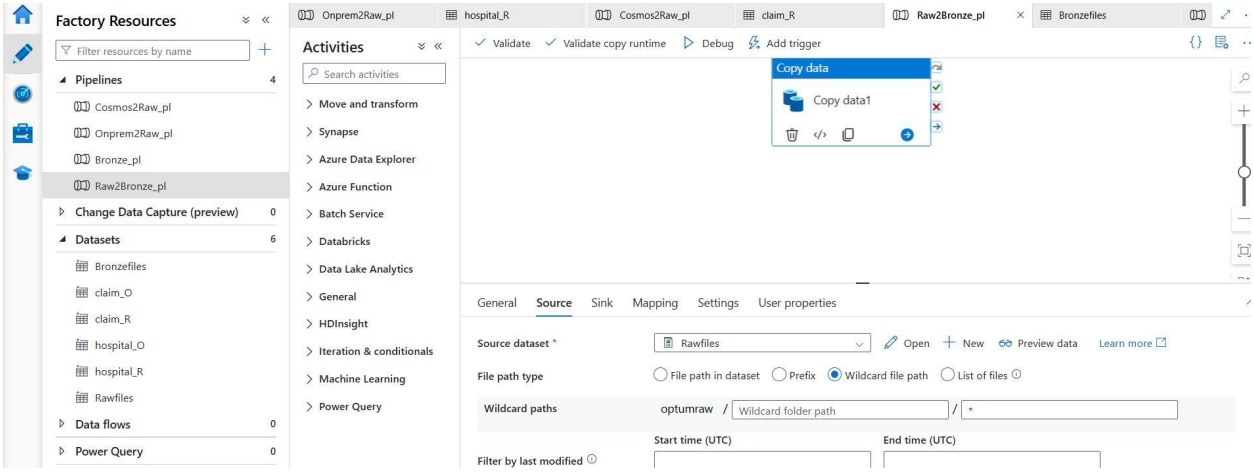
- Source : Blob, give '*' in wildcardpaths(to get all files from the container)

3.In datasets, select blob2adf_ls link service.

- Sink : Adls→Medallion→Bronze.

4.In datasets, select adls2adf_ls link service.

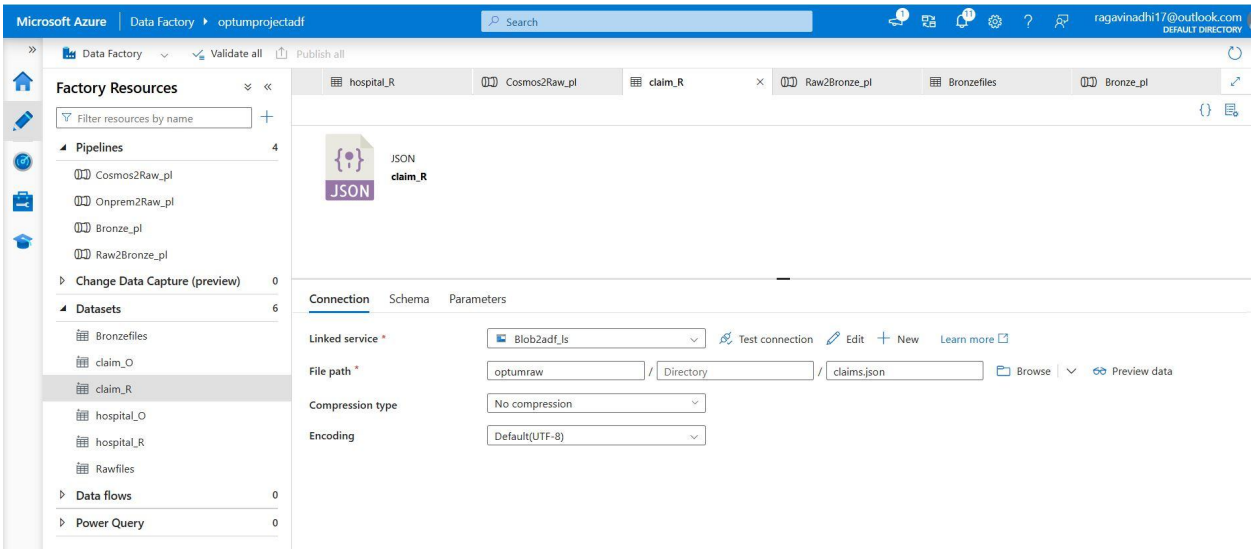
5.Trigger it and you will see the files in Bronze layer.

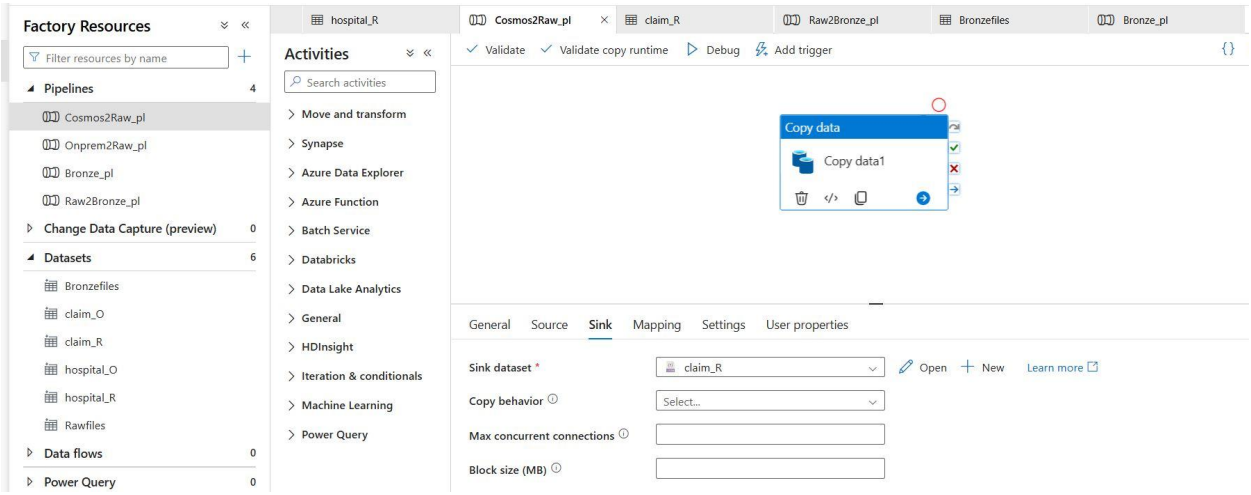


2.Cosmos2Bronze_PL:

Data move: Blob to ADLS

1. Create Pipeline in adf
2. Copy Activity
   - Source : Cosmosdb.
3. In datasets, select cosmosdb2adf_ls link service.
   - Sink : Adls→Medallion→Bronze
4. In datasets, select adls2adf_ls link service.
5. Trigger it and you will see the files in Bronze layer.



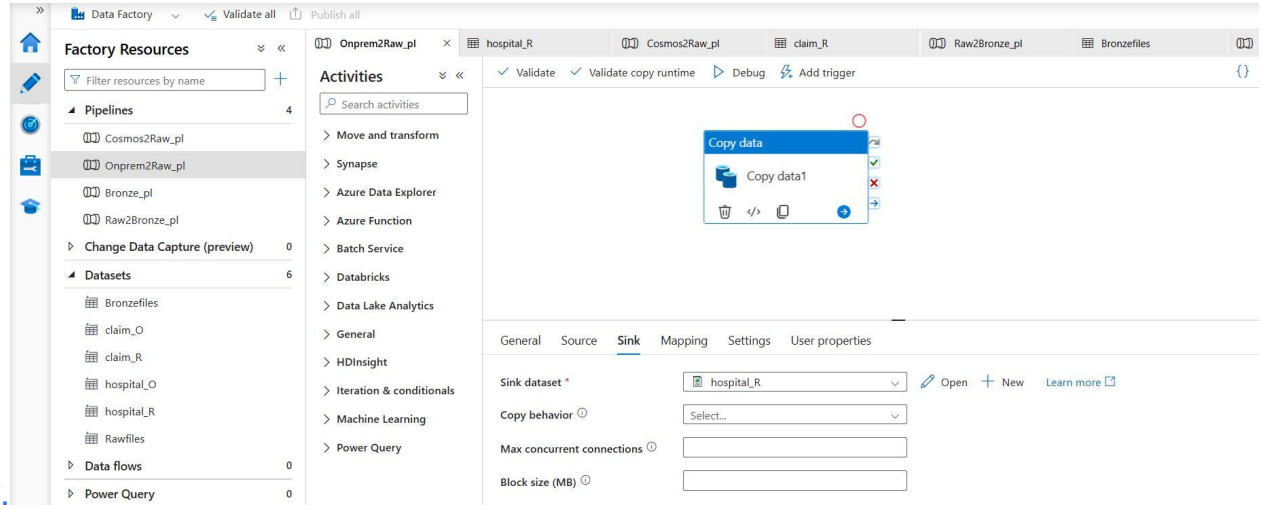


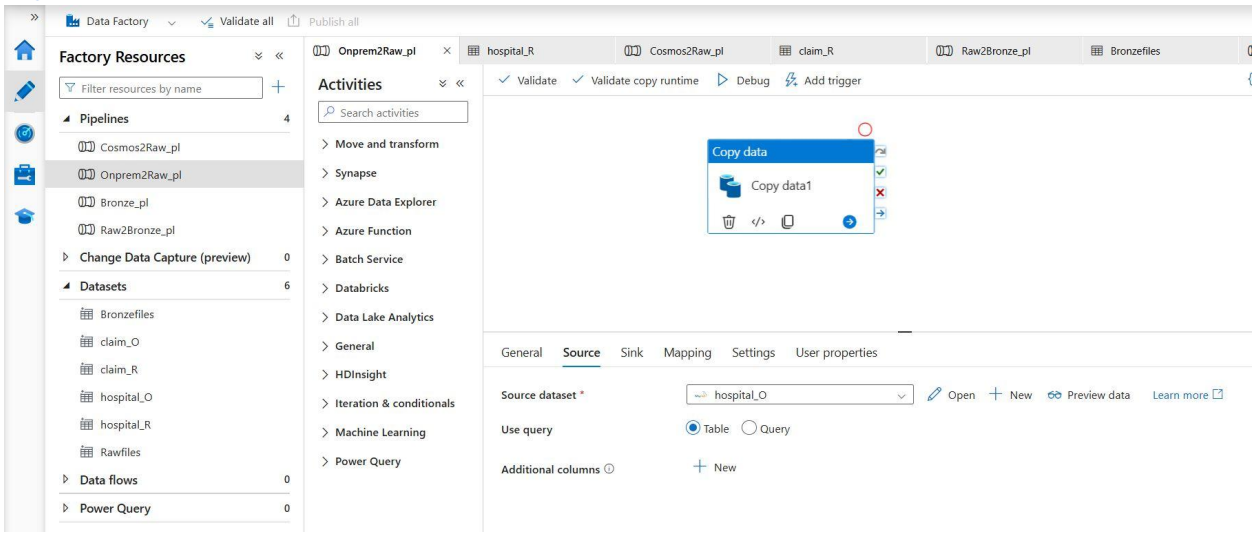
3.Omnprem2Bronze_PL:

Data move: mysql to ADLS

1. Create Pipeline in adf
2. Copy Activity
    - Source : MySQL-Onprem.
3. In datasets, select mysql2adf_ls link service.
    - Sink : Adls→Medallion→Bronze
4. In datasets, select adls2adf_ls link service.

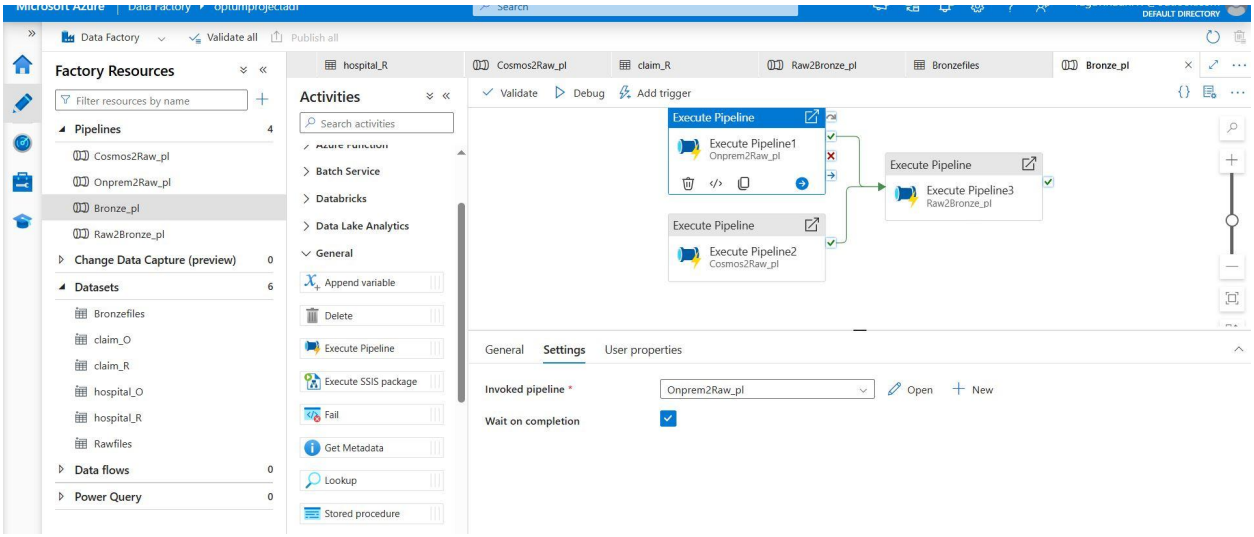
5.Trigger it and you will see the files in Bronze layer.





4.Create Bronze_PL:

Create 3 Execute Piepline, Upload all 3 pipelines which needs to go Bronze layer.



5.Bronze2Silver data:

Data move: Adls(Bronze)to adls(Silver)

5.1.Transformation in Databricks:

Step 1:

Connect Adls to Databricks using either

a.Access Token(Full Access)

b.Sas Token(Limited Access){read/write/delete etc}

c.Service Principal

2.Mount method(using above any of 3 methods)

Step 2.

1.Create Azure Key Vault

a.In Secret

b.Name : AccessKeySecret

c.Value: Password(From Blob storage)

2.And In Databricks,Give Scope Value

a.Open Azure Secret Scope Documentations

b.create secret scope in Databricks

c. in Replace <Scope-name>, give scope name.

d.Copy URL.

e.duplicate the actual databricks url and paste the above copied url.

Now,

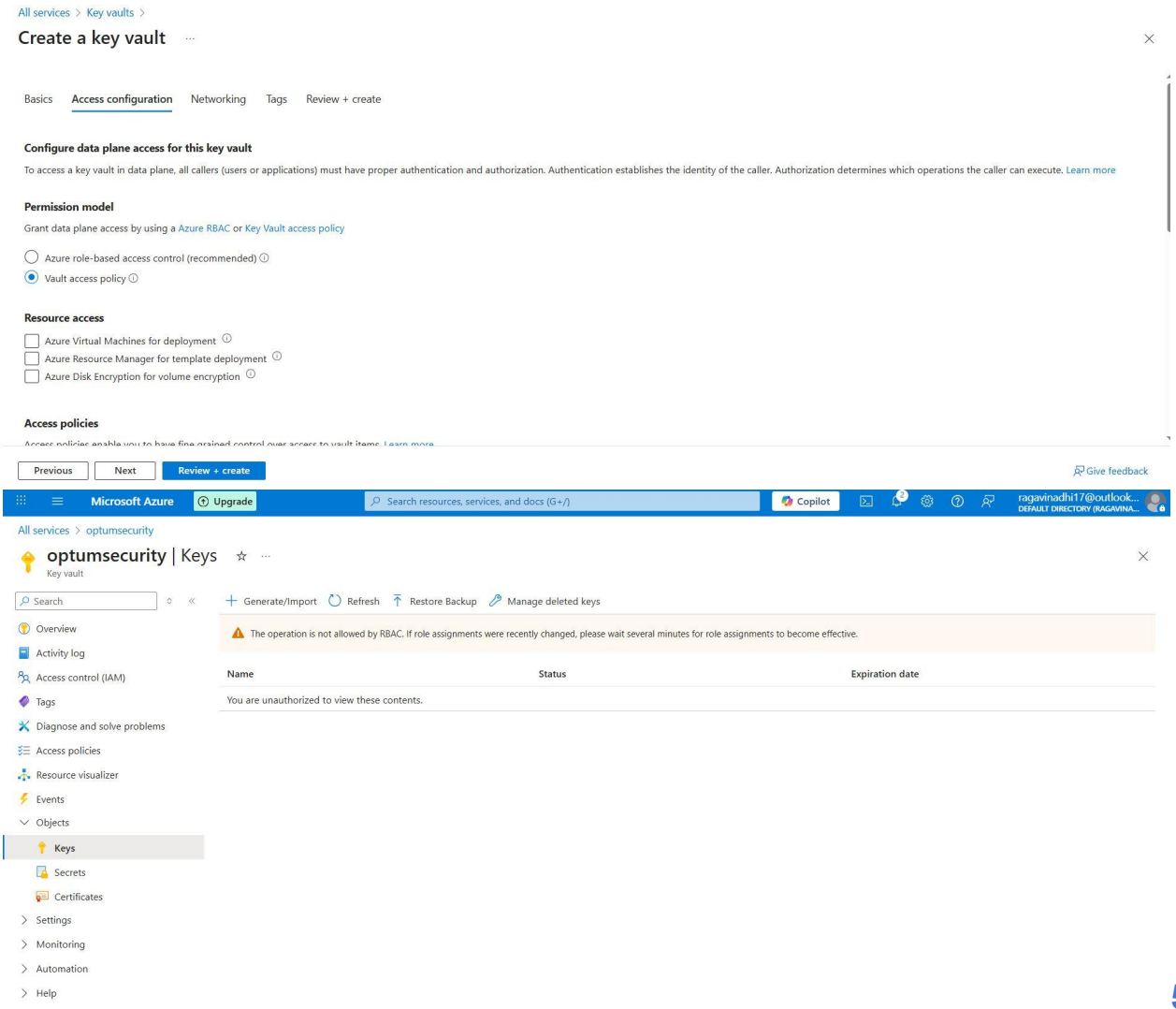
f.Create Secret screen will open.

1.Give scope name.

2.DNS name : from key vault (Overview→Properties→copy vault uri) → paste it

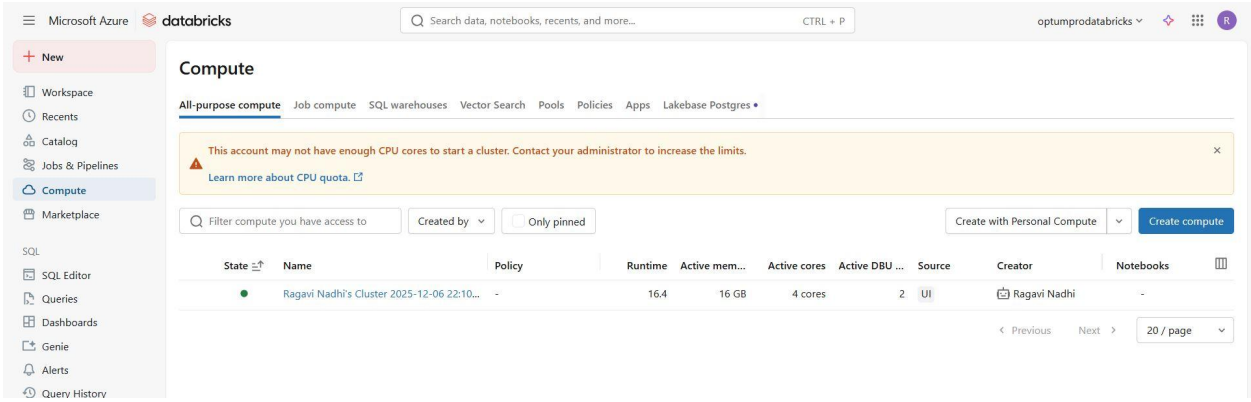
3.Resourse ID : from key vault (Properties → Copy Resource ID) → paste it

3.Copy scope name, paste it in Databricks<Scope>

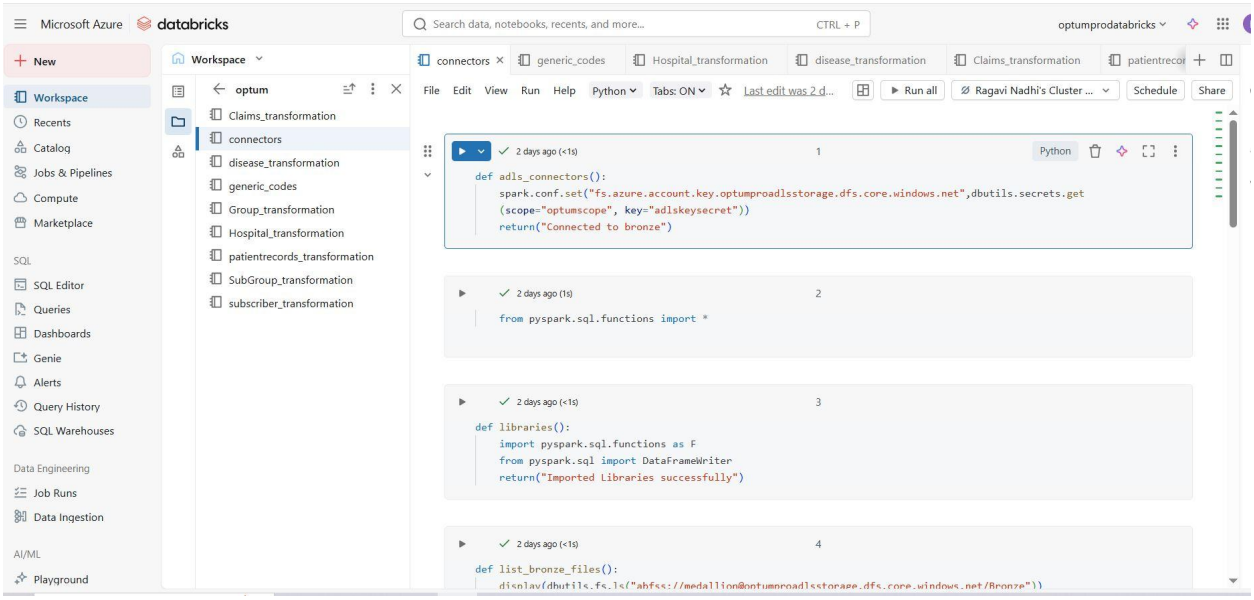


5.

2.Create Cluster in Databricks:

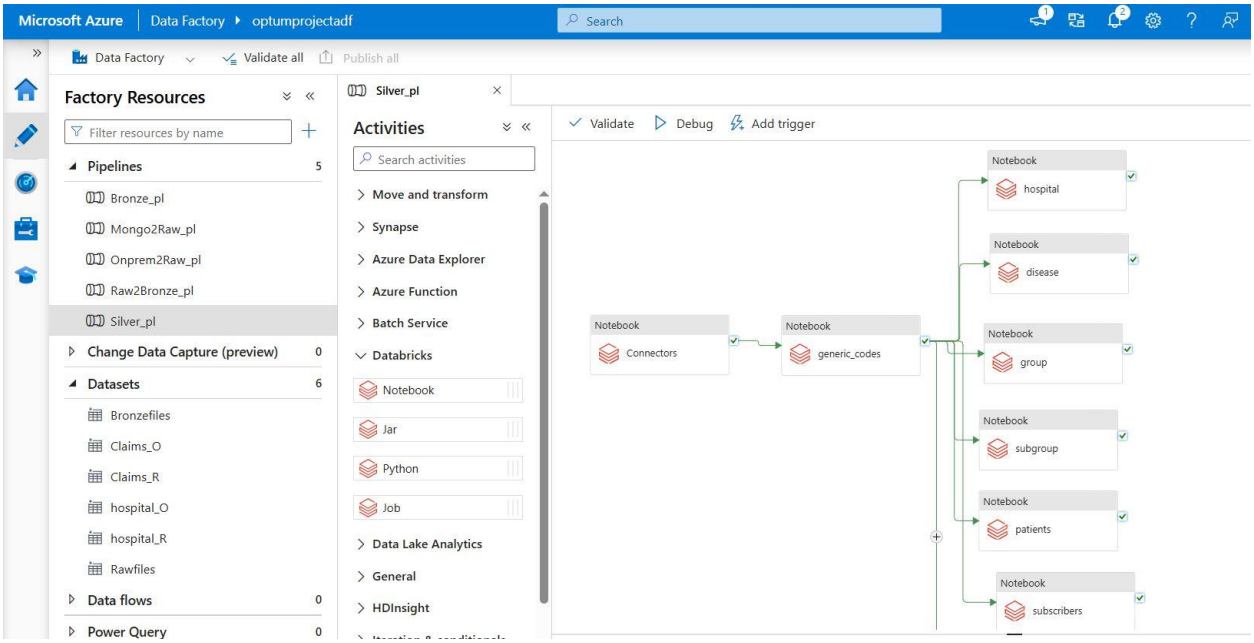


5.3.Do all the Transformations in databricks



5.4. In adf, Create new pipeline, Silver_PL:

Create Notebooks as per the datasets, invoke it and connect them based on the continuity.



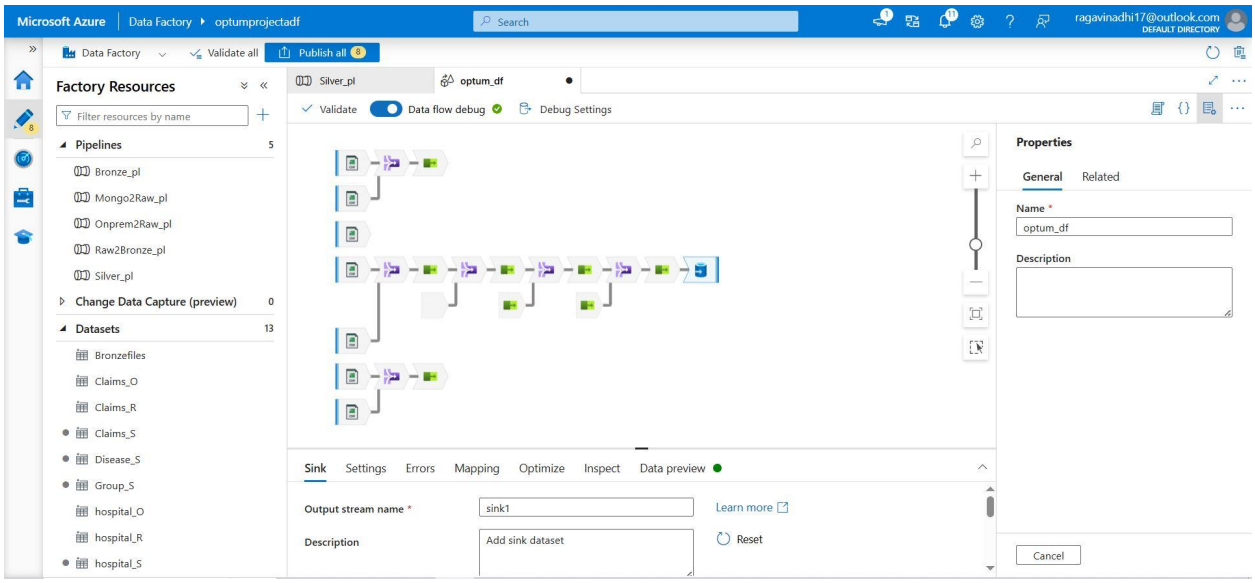
6.Silver2Gold Data:

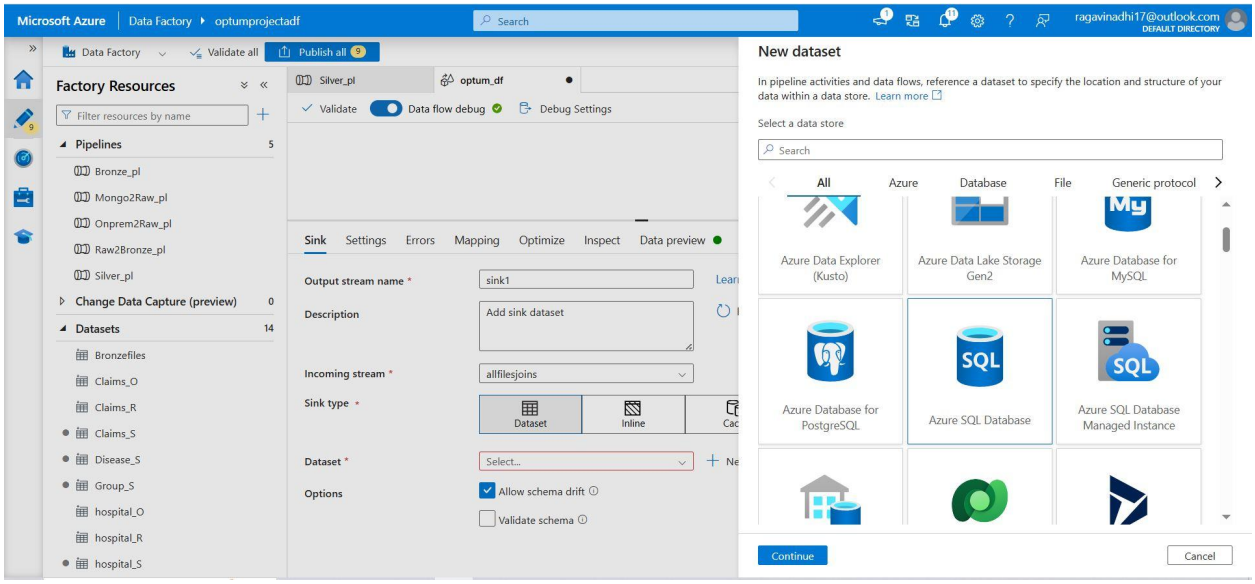
1.Create one Pipeline name as Gold_PL

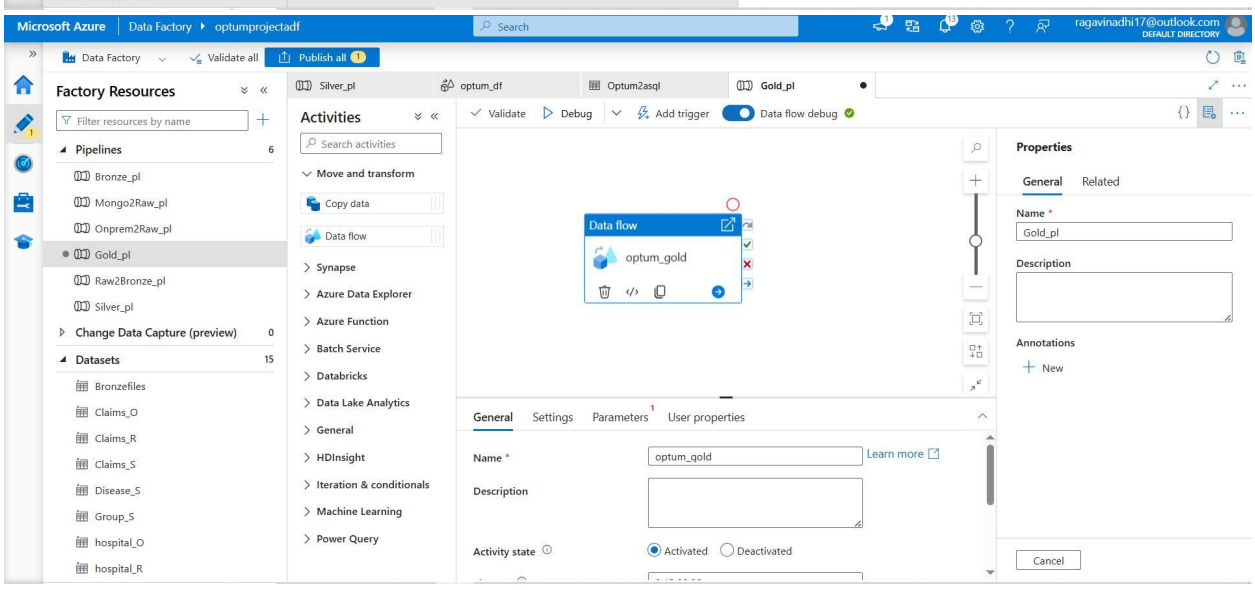
2.Create data flow name it as ‘Optum_DF’ and get Datas from silver layer

3.Connect them one by one with primaryID between the datas and finally get one single denormalised Data with required business calculations.

4.sink it to Medallion→Gold as a file and another sink in Azure SQL DB.


Microsoft Azure
Data Factory
optumprojectadf
Search
ragavinadhi17@outlook.com
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all 8
Factory Resources
Filter resources by name
Pipelines 5
Bronze_pl
Mongo2Raw_pl
Onprem2Raw_pl
Raw2Bronze_pl
Silver_pl
Change Data Capture (preview) 0
Datasets 13
Bronzefiles
Claims_O
Claims_R
Claims_S
Disease_S
Group_S
hospital_O
hospital_R
hospital_S
Silver_pl
optum_df
Validate
Data flow debug
Debug Settings
Properties
General
Related
Name *
optum_df
Description
Sink
Settings
Errors
Mapping
Optimize
Inspect
Data preview
Output stream name *
sink1
Learn more
Description
Add sink dataset
Reset
Cancel


Microsoft Azure
Data Factory
optumprojectadf
Search
ragavinadhi17@outlook.com
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all 9
Factory Resources
Filter resources by name
Pipelines 5
Bronze_pl
Mongo2Raw_pl
Onprem2Raw_pl
Raw2Bronze_pl
Silver_pl
Change Data Capture (preview) 0
Datasets 14
Bronzefiles
Claims_O
Claims_R
Claims_S
Disease_S
Group_S
hospital_O
hospital_R
hospital_S
Silver_pl
optum_df
Validate
Data flow debug
Debug Settings
Sink
Settings
Errors
Mapping
Optimize
Inspect
Data preview
Output stream name *
sink1
Description
Add sink dataset
Incoming stream *
allfilesjoins
Sink type *
Dataset
Inline
Dataset *
Select...
Options
Allow schema drift
Validate schema
New dataset
In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more
Select a data store
Search
All
Azure
Database
File
Generic protocol
Azure Data Explorer (Kusto)
Azure Data Lake Storage Gen2
Azure Database for MySQL
Azure Database for PostgreSQL
Azure SQL Database
Azure SQL Database Managed Instance
Continue
Cancel


Microsoft Azure
Data Factory
optumprojectadf
Search
ragavinadhi17@outlook.com
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all 1
Factory Resources
Filter resources by name
Pipelines 6
Bronze_pl
Mongo2Raw_pl
Onprem2Raw_pl
Gold_pl
Raw2Bronze_pl
Silver_pl
Change Data Capture (preview) 0
Datasets 15
Bronzefiles
Claims_O
Claims_R
Claims_S
Disease_S
Group_S
hospital_O
hospital_R
Silver_pl
optum_df
Optum2asql
Gold_pl
Validate
Debug
Add trigger
Data flow debug
Activities
Search activities
Move and transform
Copy data
Data flow
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
HDInsight
Iteration & conditionals
Machine Learning
Power Query
Data flow
optum_gold
Properties
General
Related
Name *
Gold_pl
Description
Annotations
New
General
Settings
Parameters
User properties
Name *
optum_gold
Learn more
Description
Activity state
Activated
Deactivated
Cancel

7.Optum_PL:

1.Create one Pipeline ‘Optum_PL’

2.Create 3 execute_pipeline and invoke bronze_pl, silver_pl and gold_pl in here

3.Trigger ‘Optum_PL’, it will run all pipelines and we will see the final Output in AzureSqlDB and Gold layer in ADLS.